AMENDMENTS TO THE CLAIMS

Please amend Claims 2, 10, 16, 23, 31, 37, and 44 as follows. Please add new Claims 50—54. Claims 2-7, 9-12, 14-16, 23-29, 31-35, 37, and 39-54 are pending.

1. (Canceled)

2. (Currently Amended) A method for providing data transfers between a processor and a component, the method comprising:

routing requests originating from a component to a processor through a target controller and handling requests originating from the processor by;

buffering a first address with a first address buffer and a second address with a second address buffer, the first and second address buffers being in communication with ~~a~~ the processor and ~~a~~ the component, wherein the processor operates at a different speed than the component;

buffering a first data value with a first bi-directional data buffer and buffering a second data value with a second bi-directional data buffer, the first and second bi-directional data buffers being in communication with the processor and the component, wherein the first and second address buffers are separate from the first and second data buffers;

monitoring the first and second address buffers and the first and second data buffers to determine when the first address and data buffers and the second address and data buffers have completed a task and are available for a further task;

controlling the first address buffer and the first bi-directional data buffer as a matched pair such that the first address held in the first address buffer corresponds to the first data value held in the first bi-directional data buffer;

controlling the second address buffer and the second bi-directional data buffer as a matched pair such that the second address held in the second address buffer corresponds to the second data value held in the second bi-directional data buffer;

reading status information from the first address buffer to determine a priority status of the first data value;

reading status information from the second address buffer to determine the priority status of the second data value; and

controlling the order of bi-directional data flow through the first and second bi-directional data buffers such that data flows concurrently between the processor and the component while the processor is processing other instructions and, wherein controlling the order of the bi-directional data flow through the first and second bi-directional data buffers is variable and based on the priority status of the first and second data values.

3. (Previously Presented) The method of Claim 2, wherein the first and second bi-directional data buffers are in communication with the processor via a bus.

4. (Previously Presented) The method of Claim 3, wherein the first and second bi-directional data buffers are in communication with the bus via a bus master controller and a bus slave controller.

5. (Previously Presented) The method of Claim 2, wherein the first address buffer further comprises status bits.

6. (Previously Presented) The method of Claim 5, wherein the status bits relate to the type of request being made by the processor.

7. (Previously Presented) The method of Claim 2, wherein said controlling the first address buffer and the first bi-directional data buffer as a matched pair is performed with pointers.

8. (Cancelled)

9. (Previously Presented) The method of Claim 2, wherein said act of controlling bi-directional data flow is performed with at least one input data arbiter.

10. (Currently Amended) A method for controlling data transfers between a processor and a component, the method comprising:

routing requests originating from a component to a processor through a target controller;

buffering with a plurality of address buffers ~~address~~ requests originating from ~~a~~ the processor to the component, wherein the processor operates at a different speed than the component;

storing status information in each of the plurality of address buffers, the status information determining the priority status of data transfers associated with the address requests;

monitoring the plurality of address buffers and the first to determine when address buffers have completed a task and are available for a further task;

bi-directionally buffering with a plurality of bi-directional data buffers data transfers between the processor and the component, wherein said data transfers can be performed out of a previously defined order based on the priority status of each of the data transfers and such that data transfers can be performed concurrently while the processor is processing other instructions; and

controlling said buffering address requests and said bi-directionally buffering through said plurality of bi-directional data buffers such that each of the buffered data transfers relates to an address held in one of the plurality of address buffers.

11. (Previously Presented) The method of Claim 10, additionally comprising indicating which of the plurality of bi-directional data buffers is available to accept new data.

12. (Previously Presented) The method of Claim 11, wherein said act of indicating is performed with reference pointers.

13. (Cancelled)

14. (Previously Presented) The method of Claim 10, wherein said act of buffering address requests includes the use of an input arbiter and an output arbiter.

15. (Previously Presented) The method of Claim 10, wherein said act of bi-directionally buffering is performed with an input arbiter and an output arbiter.

16. (Currently Amended) The method of Claim 10, wherein the plurality of address buffers comprises at least three address buffers and wherein the plurality of bi-directional data buffers comprises at least three bi-directional data buffers and wherein each address buffer is matched as a pair with a corresponding data buffer.

17.–22. (Cancelled).

23. (Currently Amended) A method for transferring addresses and data through a bi-directional buffer, the method comprising:

routing requests of a first component originating from a second component through a target controller and handling requests originating from the first component by;

storing a first address in a first buffer in communication with the first component and the second component, the first buffer comprising status bits;

storing first data in a second bi-directional buffer matched with said first buffer so that the first address stored in the first buffer is related to the first data stored in the second bi-directional buffer;

storing a second address in a third buffer in communication with the first component and the second component, the third buffer comprising status bits;

storing second data in a fourth bi-directional buffer matched with said third buffer so that the second address stored in the third buffer is related to the second data stored in the fourth bi-directional buffer;

monitoring the first and second address buffers and the first and second data buffers to determine when the first address and data buffers and the second address and data buffers have completed a task and are available for a further task;

reading the status bits of the first buffer to determine a first priority value of the first data;

reading the status bits of the third buffer to determine a second priority value of the second data; and

controlling the order of bi-directional data flow of the first data and the second data through said second and fourth bi-directional buffers in a variable manner based at least in part on said first and second priority values and controlling the bi-directional data flow such that data flows concurrently with processing by the second component of other instructions.

24. (Previously Presented) The method of Claim 23, wherein the status bits comprise transfer type bits indicative of the status of an address transfer from the first component to the first buffer.

25. (Previously Presented) The method of Claim 23, wherein the status bits comprise transfer type bits indicative of the status of a data transfer from the first component to the second buffer.

26. (Previously Presented) The method of Claim 23, wherein the first component comprises a memory.

27. (Previously Presented) The method of Claim 23, wherein the first component comprises a processor.

28. (Previously Presented) The method of Claim 27, wherein the first buffer is in communication with the processor via a bus.

29. (Previously Presented) The method of Claim 28, wherein the first buffer is in communication with the bus via a bus master controller and a bus slave controller.

30. (Cancelled)

31. (Currently Amended) A method for transferring data between a processor and a component utilizing a plurality of address buffers and a plurality of data buffers, the method comprising:

when a request originates from the processor, receiving a data read or write request including an associated address from the processor;

determining whether at least one of a plurality of address buffers and an associated bi-directional data buffer are available;

storing the associated address in the at least one address buffer;

storing status information indicative of a priority of the data request in the at least one address buffer;

~~receiving~~ buffering data identified by the associated address ~~from the component~~ with the bi-directional data buffer; and

ordering, based on the priority of the data request, the transmission of the data from the bi-directional data buffer to the processor and such that data flows bi-directionally and concurrently with processing by the processor of other instructions; and

when a request originates from the component, routing the request through a target controller.

32. (Previously Presented) The method of Claim 31, additionally comprising receiving the address into the at least one address buffer while data is being read from the bi-directional data buffer.

33. (Previously Presented) The method of Claim 31, wherein the at least one address buffer and the bi-directional data buffer are in communication with the processor via a bus.

34. (Previously Presented) The method of Claim 33, wherein the at least one address buffer and the bi-directional data buffer are in communication with the bus via a bus master controller and a bus slave controller.

35. (Previously Presented) The method of Claim 31, wherein the bi-directional data buffer and the at least one address buffer are associated with each other through the use of pointers.

36. (Cancelled)

37. (Currently Amended) An apparatus for controlling data transfers between a processor and a component, the apparatus comprising:

means for buffering address requests from a processor to a component;

means for bi-directionally buffering data transfers between the processor and the component;

means for storing status information indicative of a priority status of the buffered data transfer;

means for controlling the means for buffering and the means for bi-directionally buffering so that each of the buffered data transfers relates to an address held in the means for buffering, wherein the means for controlling further coordinates an order of said data transfers based at least on the priority status of each buffered data transfer and such that data flows bi-directionally and concurrently with processing by the processor of other instructions; and

means for routing requests from the component to the processor.

38. (Cancelled).

39. (Previously Presented) The apparatus of Claim 37, wherein the means for buffering includes a plurality of address buffers.

40. (Previously Presented) The apparatus of Claim 37, wherein the means for bi-directionally buffering includes a plurality of data buffers.

41. (Previously Presented) The apparatus of Claim 37, wherein the means for buffering includes an input arbiter and an output arbiter.

42. (Previously Presented) The apparatus of Claim 37, wherein the means for bi-directionally buffering includes an input arbiter and an output arbiter.

43. (Previously Presented) The method of Claim 23, additionally comprising providing signals with an arbiter in communication with said status bits to grant access to the first buffer

and to the second buffer such that a third address can be written to the first buffer while data is read from the second buffer.

44. (Currently Amended) A buffer allocation system for managing data flow between components of a computer, the system comprising:

an address buffer module configured to handle address requests between a first component and a second component for requests originating from the first component, the address buffer module comprising:

a plurality of address buffers each in communication with the first and second components, each address buffer comprising status information indicative of a priority status;

an input address arbiter configured to direct the address requests to the plurality of address buffers; and

an output address arbiter configured to send the address requests from the plurality of address buffers to the second component; and

a data buffer module configured to control an order of bi-directional flow of data therethrough based on the priority status of the data and such that data flows bi-directionally and concurrently with processing by the first component of other instructions, the data buffer module comprising:

a plurality of bi-directional data buffers each in communication with the first and second components;

an input data arbiter configured to direct data to the plurality of bi-directional data buffers; and

an output data arbiter configured to direct data output from the plurality of bi-directional data buffers and

a target controller configured to route requests between the first and second components for requests originating from the second component.

45. (Previously Presented) The system of Claim 44, wherein the plurality of address buffers comprises a first address buffer and the plurality of bi-directional data buffers comprises a first bi-directional data buffer, wherein an address held in the first address buffer corresponds only to a data value held in the first bi-directional data buffer.

46. (Previously Presented) The system of Claim 45, wherein the plurality of address buffers comprises a second address buffer and the plurality of bi-directional data buffers comprises a second bi-directional data buffer, wherein a second address held in the second address buffer corresponds only to a second data value held in the second bi-directional data buffer.

47. (Previously Presented) The system of Claim 44, wherein the address buffer module further comprises a plurality of output multiplexers coupled to the output address arbiter and the plurality of address buffers.

48. (Previously Presented) The system of Claim 44, wherein the data buffer module further comprises a plurality of output multiplexers coupled to the input data arbiter and the plurality of bi-directional data buffers.

49. (Previously Presented) The apparatus of Claim 37, wherein the means for routing requests from the component to the processor comprises a PCI target controller.

Please add the following new Claims.

50. (New) The method of Claim 2, wherein the data flowing concurrently comprises one or more of:

the processor writing data to the first data buffer at the same time that data is being read from a PCI bus into the second data buffer;

a deferred data read from the first data buffer occurring concurrently with a data read from a PCI bus to the second data buffer; and

performing a deferred data read from the first data buffer to the processor at the same time as performing a data write operation from the second data buffer to a PCI bus.

51. (New) The method of Claim 10, wherein the concurrent data transfer comprises one or more of:

the processor writing data to a first data buffer at the same time that data is being read from a PCI bus into a second data buffer;

a deferred data read from a first data buffer occurring concurrently with a data read from a PCI bus to a second data buffer; and

performing a deferred data read from a first data buffer to the processor at the same time as performing a data write operation from a second data buffer to a PCI bus.

52. (New) The method of Claim 23, wherein the data flowing concurrently comprises one or more of:

the first component writing data to the second buffer at the same time that data is being read from a PCI bus into the fourth buffer;

a deferred data read from the second buffer occurring concurrently with a data read from a PCI bus to the fourth buffer; and

performing a deferred data read from the second buffer to the first component at the same time as performing a data write operation from the fourth buffer to a PCI bus.

53. (New) The method of Claim 31, wherein the data flowing concurrently comprises one or more of:

the processor writing data to a first data buffer at the same time that data is being read from a PCI bus into a second data buffer;

a deferred data read from a first data buffer occurring concurrently with a data read from a PCI bus to a second data buffer; and

performing a deferred data read from a first data buffer to the processor at the same time as performing a data write operation from a second data buffer to a PCI bus.

54. (New) The system of Claim 44, wherein the data flowing concurrently comprises one or more of:

the first component writing data to a first data buffer at the same time that data is being read from a PCI bus into a second data buffer;

a deferred data read from a first data buffer occurring concurrently with a data read from a PCI bus to a second data buffer; and

performing a deferred data read from a first data buffer to the first component at the same time as performing a data write operation from a second data buffer to a PCI bus.